Computing a Partial Mapping by a P System: Design and Verification

Antonio PÉREZ-JIMÉNEZ, Mario J. PÉREZ–JIMÉNEZ, Fernando SANCHO–CAPARRINI
Dpt. Computer Science and Artificial Intelligence
University of Seville
Avda. Reina Mercedes s/n. 41012, Seville, Spain
E-mail: {Antonio.Perez, Mario.Perez, Fernando.Sancho}@cs.us.es

Abstract. Computing with membranes is a new computability model and it is basically a non imperative and procedural model. For that reason it is very hard to establish the verification of the P systems. In this paper a computing P system (according to the definition given in section 2) which computes the set $\{1^2, 2^2, \ldots, n^2\}$ for a given $n \geq 1$, is presented. A formalization of its syntax is given and the verification of this computing P system is established through the characterization of its successful computations.

1 Introduction

In October 1998, Gheorghe Păun ([1]) introduces a new computability model, of a distributed parallel type, based on the notion of *membrane structure*. This model, called *transition P-systems*, start from the observation that the processes which take place in the complex structure of a living cell can be considered *computations*.

The *membrane structure* of a P system is a hierarchical arrangement of membranes (understood as vesicles in a space), embedded in a *skin membrane* that separates the system from the environment. When a membrane has not any membrane inside, it is called *elementary*. Each membrane encloses a space between it and the membranes directly included in it (if any). This space (the *region* of the membrane) can contain a multiset (a set where the elements can be repeated) of objects (represented by symbols of a given alphabet) and a set of (evolution) rules for them. Each membrane defines an unique region; that is, each region is delimited (*from the outside*) by an unique membrane.

In [1], Gh. Păun illustrates the way of working of this new model giving an example of a transition P system *generating* exactly all squares of natural numbers greater or equal to 1. In [4] a formal verification of that P system has been given. In this paper we present a computing P system Π (according to the definition given in Section 2) such that for every natural number $n \geq 1$, the P system Π with input n returns the set of squares $\{1^2, 2^2, \ldots, n^2\}$.

The paper is organized as follows: Section 2 briefly presents some basic concepts about computing transition P systems. Section 3 gives a computing P system Π, formalizing its syntaxis according to [3]. In section 4 some properties of this P system are studied in order to characterize the successful computations of it. In Section 5, this P system is shown to be able to compute the partial function $f : \mathbf{N} - \rightarrow P(\mathbf{N})$ defined as follows:

$$f(n) = \begin{cases} \uparrow & \text{if } n = 0 \\ \{1^2, 2^2, \ldots, n^2\} & \text{if } n \neq 0 \end{cases}$$

2 Preliminaries about transition P systems

Following [3], a *membrane structure* is a rooted tree, where the nodes are called *membranes*, the root is called *skin*, and the leaves are called *elementary membranes.* A *cell* (or *super-cell*) over an alphabet, A, is a pair (μ, M), where $\mu = (V(\mu), E(\mu))$ is a membrane structure, and M is an application, $M : V(\mu) \longrightarrow \mathbf{M}(A)$ (the set of multisets over A). Over the elements of $E(\mu)$ a directionality, $E^*(\mu)$ can be considered induced from the node root.

Let (μ, M) be a cell over an alphabet, A. Let $x \in V(\mu)$. An *evolution rule* associated to x is a 3-tuple $r = (\vec{d_r}, \vec{v}_r, \delta_r)$ where

- $\vec{d_r}$ is a multiset over A.
- $\vec{v}_r$ is a function from $V(\mu) \cup \{here, out\}$ to $\mathbf{M}(A)$ where $here, out \notin V(\mu)$ ($here \neq out$).
- $\delta_r \in \{\neg\delta, \delta\}$, with $\neg\delta, \delta \notin A$ ($\neg\delta \neq \delta$).

A *collection R of evolution rules* associated to C is a function with domain $V(\mu)$ such that for every membrane $x \in V(\mu)$, $R_x = \{r_1^x, \ldots, r_{s_x}^x\}$ is a finite set (possibly empty) of (evolution) rules associated to x. A *priority relation over R* is a function, ρ, with domain $V(\mu)$ such that for every membrane $x \in V(\mu)$, ρ_x is a strict partial order over R_x (possibly empty).

A *transition P-system* is a 4-tuple $\Pi = (A, C_0, \mathcal{R}, i_0)$, where:

- A is a non-empty finite set (usually called base alphabet).
- $C_0 = (\mu_0, M_0)$ is a cell over A.
- $\mathcal{R}$ is an ordered pair (R, ρ) where R is a collection of (evolution) rules associated to C_0, and ρ is a priority relation over R.
- i_0 is a node of μ_0, which specifies the output membrane of Π.

A *configuration*, C, of a P system, $\Pi = (A, C_0, \mathcal{R}, i_0)$ with $C_0 = (\mu_0, M_0)$, is a cell $C = (\mu, M)$ over A, where $V(\mu) \subseteq V(\mu_0)$, and μ has the same root as μ_0. The configuration C_0 will be called the *initial configuration of* Π. Let $x \in V(\mu_0)$, we say that the (evolution) rule $r \in R_x$ is *semi-applicable* to C if:

- The membrane associated to node x exists in C, that is, $x \in V(\mu)$.
- Dissolution is not allowed in root node, that is, if x is the root node of μ, then $\delta_r = \neg\delta$.
- The membrane associated to x has all the necessary objects to apply the rule, that is, $\vec{d_r} \leq M(x)$.

- Nodes where the rule tries to send objects (by means of in_y) are children of x, that is, $\forall\, y \in V(\mu)(\vec{v}_r(y) \neq \vec{0} \rightarrow (x,y) \in E^*(\mu))$

We say that the rule $r \in R_x$ is *applicable* to C, if it is semi-applicable to C and there is no semi-applicable rules in R_x with higher priority. That is to say:

$$\neg\exists\, r'\ (r' \in R_x \wedge \rho_x(r', r) \wedge r' \text{ semi-applicable to } C)$$

We will say that $\vec{p} \in \mathbf{N}^{\mathbf{N}}$ is an *applicability vector* over $x \in V(\mu)$ for C, and we will denote it as $\vec{p} \in \mathbf{Ap}(x, C)$, if:

- The node is still alive, that is, $\vec{p} \neq \vec{0} \Rightarrow x \in V(\mu)$.
- It has correct size, that is, $\forall\, j\ (j > s_x \rightarrow \vec{p}(j) = 0)$, (where s_x is the number of rules associated to x).
- Every rule can be applied as many times as the vector $\vec{p}$ indicates, that is,
$$\forall\, j\ (1 \leq j \leq s_x \rightarrow \vec{p}(j) \leq N_{Ap}(r_j^x, C, x))$$
- All the rules can be applied simultaneously, that is, $\displaystyle\sum_{j=1}^{s_x} \vec{p}(j) \otimes \vec{d}_{r_j^x} \leq M(x)$.
- It is maximal, that is, $\neg\exists\, \vec{v} \in \mathbf{N}^{\mathbf{N}}\ (\vec{p} < \vec{v} \wedge \vec{v} \in \mathbf{Ap}(x, C))$.

We will say that $P : V(\mu_0) \longrightarrow \mathbf{N}^{\mathbf{N}}$ is an *applicability matrix* over C, denoted $P \in \mathbf{M_{Ap}}(C)$, if for every $x \in V(\mu_0)$ we have that $P(x) \in \mathbf{Ap}(x, C)$. We define $\Delta(P, C) = \{x :\ x \in V(\mu) \wedge \exists\, j\ (1 \leq j \leq s_x \wedge P_x(j) \neq 0 \wedge \delta_{r_j^x} = \delta)\}$. For each node $x \in V(\mu)$, we define the *donors* of x for C in the application of P as follows:

$$Don(x, P, C) = \begin{cases} \emptyset & ,\text{if } x \in \Delta(P, C) \\ \{y \in V(\mu) : y \in \Delta(P, C) \wedge x \rightsquigarrow_\mu y\ \wedge & \\ \quad \wedge\ \forall\, z \in V(\mu)(x \rightsquigarrow_\mu z \rightsquigarrow_\mu y \rightarrow z \in \Delta(P, C))\} & ,\text{if } x \notin \Delta(P, C) \end{cases}$$

We define the *execution* of P over C, denoted $P(C)$, as the configuration of Π, $C' = (\mu', M')$, where:

- μ' is the rooted tree obtained from μ by means of:
  - $V(\mu') = V(\mu) - \Delta(P, C)$
  - If $x, y \in V(\mu')$, then:
$$\begin{aligned}(x,y) \in E^*(\mu') \Leftrightarrow\ & \exists\, x_0, \ldots, x_n \in V(\mu)(x_1, \ldots, x_{n-1} \in \Delta(P,C) \wedge x_0 = x \wedge \\ & x_n = y \wedge \forall\, i\ (0 \leq i < n \rightarrow (x_i, x_{i+1}) \in E^*(\mu)))\end{aligned}$$
- $M'(x) = \begin{cases} M''(x) \cup \displaystyle\bigcup_{y \in Don(x,P,C)} M''(y) & ,\ \text{if } x \notin \Delta(P, C) \\ \emptyset & ,\ \text{if } x \in \Delta(P, C) \end{cases}$

We will say that a configuration C_1 of a P system Π yields a configuration C_2 by a *transition in one step* of Π, denoted $C_1 \Rightarrow_\Pi C_2$, if there exists a non–zero applicability matrix over C_1, P, such that $P(C_1) = C_2$.

The *computation tree of a P system* Π, denoted $\mathbf{Comp}(\Pi)$, is a rooted labeled maximal tree defined as follows: the root of the tree is the initial configuration, C_0, of Π. The children of a node are the configurations that follow in one step of transition. Nodes and edges are labeled by configurations and applicability matrices, respectively, in such way that two labeled nodes C, C' are adjacent in $\mathbf{Comp}(\Pi)$, by means an edge labeled with P, if and only if $P \in \mathbf{M_{Ap}}(C) - \{\mathbf{0}\} \wedge C' = P(C)$. The maximal branches of $\mathbf{Comp}(\Pi)$ will be called *computations* of Π. We will say that a computation of Π *halts* if it is a finite branch. The configurations verifying $\mathbf{M_{Ap}}(C) = \{\mathbf{0}\}$ will be called *halting configurations*.

We say that a computation $C_0 \Rightarrow_\Pi C_1 \Rightarrow_\Pi \ldots \Rightarrow_\Pi C_n$, where $C_n = (\mu_n, M_n)$, of a P system $\Pi = (A, C_0, \mathcal{R}, i_0)$ is *successful* if it halts and i_0 is a leaf of the rooted tree μ_n (also, we will say that C_n is *successful*). We will denote as $S(\Pi)$ the set of the successful configurations of Π. The *numerical output* of a successful computation, $\mathcal{C}$, is $O(\mathcal{C}) = |M_{C_n}(i_0)|$ where C_n is the last configuration of $\mathcal{C}$. The output of a P system Π is $O(\Pi) = \{|M_{C_n}(i_0)| : \ C_n \in S(\Pi)\} = \{O(\mathcal{C}) : \ \mathcal{C}$ is a successful computation of $\Pi\}$.

A *computing* P system of order k is a 6–tuple $\Pi = (A, B, C_0, \mathcal{R}, i_0, j_0)$ such that

- The 4-tuple $\Pi' = (A, C_0, \mathcal{R}, i_0)$ is a P system.
- $B = (a_1, \ldots, a_k)$ is a k-tuple of elements of A, pairwise distinct (distinguished elements of the base alphabet to encode the input data).
- j_0 is a node of μ_0 which specifies the input membrane of Π.

A computing P system $\Pi = (A, B, C_0, \mathcal{R}, i_0, j_0)$ of order k *computes* the partial map $f : \mathbf{N}^k - \rightarrow P(\mathbf{N})$ if for every $(n_1, \ldots, n_k) \in \mathbf{N}^k$ we have that

- $\forall t \ (1 \leq t \leq k \rightarrow (M_{C_0}(j_0))(a_t) = n_t)$, with $B = (a_1, \ldots, a_k)$; that is, the input membrane of the initial configuration of the P system encodes $(n_1, \ldots, n_k)$.
- For every $(n_1, \ldots, n_k) \in \mathbf{N}^k$, $f(n_1, \ldots, n_k)$ halts if and only if the P system with input $(n_1, \ldots, n_k)$ halts (there are successful configurations) and, also, $f(n_1, \ldots, n_k) = \{|M_C(i_0)| : C \in S(\Pi') \wedge C = (\mu_C, M_C)\}$.

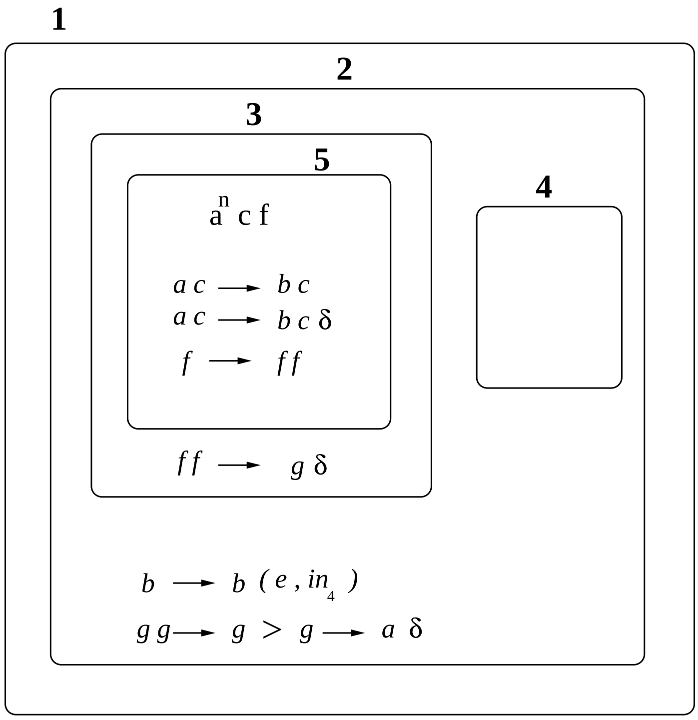
Note that a partial map, $f : \mathbf{N}^k - \rightarrow P(\mathbf{N})$, is not computed by one P system but by a collection of *similar* P systems, where, in their definition, only the multiset of objects of the input membrane differs.

3 A computing P system that computes squares

In this section we present a computing P system, Π, of order 1 that computes the partial function $f : \mathbf{N} - \rightarrow P(\mathbf{N})$ defined as follows:

$$f(n) = \begin{cases} \uparrow & \text{if } n = 0 \\ \{1^2, 2^2, \ldots, n^2\} & \text{if } n \neq 0 \end{cases}$$

The computing P system Π we present here can be graphically described as follows:



P system Π with input n

where membrane 4 is the output one, and $n \in \mathbf{N}$ is an input data.

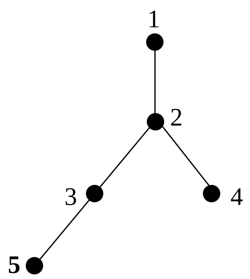
Now, according to the paper [3], we formalize the syntax of Π.

The computing P system Π is a 6–tuple $(A, B, C_0, \mathcal{R}, i_0, j_0)$, where:

(a) The base alphabet is $A = \{a, b, c, e, f, g\}$ and $B = (a)$.

(b) The initial configuration, $C_0 = (\mu_0, M_0)$, is defined as follows:

$\mu_0 = (1, ((1, 2), (2, 1, 3, 4), (3, 2, 5), (4, 2), (5, 3)))$. That is, μ_0 is the membrane structure given by means of the following rooted tree (with nodes labeled by natural numbers):



M_0 is the application from $\{1, 2, 3, 4, 5\}$ to $\mathbf{M}(A)$ defined as follows: $M_0(1) = M_0(2) = M_0(3) = M_0(4) = \emptyset$ and $M_0(5) = \{a^n f\}$, for the input data $n \in \mathbf{N}$.

(c) $\mathcal{R} = (R, \rho)$, where:

- R is the application with the domain $\{1, 2, 3, 4, 5\}$ defined as: $R(1) = R(4) = \emptyset$, $R(2) = \{r_1^2, r_2^2, r_3^2, \}$, $R(3) = \{r_1^3\}$ and $R(5) = \{r_1^5, r_2^5, r_3^5\}$, where:

Rule	$\mathbf{d_r}$	$\mathbf{v_r}$(here)	$\mathbf{v_r}$(here)	δ
r_1^2	b	b	e	$-$
r_2^2	gg	g	$-$	$-$
r_3^2	g	a	$-$	$+$
r_1^3	ff	g	$-$	$+$
r_1^5	ac	bc	$-$	$-$
r_2^5	ac	bc	$-$	$+$
r_3^5	f	ff	$-$	$-$

- ρ is the application with domain $\{1, 2, 3, 4, 5\}$ defined as: $\rho(1) = \rho(3) = \rho(4) = \rho(5) = \emptyset$ and $\rho(2) = \{(r_2^2, r_3^2)\}$.

(d) The output membrane is $i_0 = 4$ and the input membrane is $j_0 = 5$.

Given $n \in \mathbf{N}$, we will denote by $\Pi(n)$ the P system Π with input data n.

4 Characterizing successful configurations of $\Pi(n)$

The main goal of this paper is to present a formal proof of the fact that the computing P system Π presented in the previous section actually computes the partial function $f : \mathbf{N} - \rightarrow P(\mathbf{N})$ defined as follows:

$$f(n) = \begin{cases} \uparrow & \text{if } n = 0 \\ \{1^2, 2^2, \ldots, n^2\} & \text{if } n \neq 0 \end{cases}$$

To establish the verification of $\Pi(n)$ in relation to the function $f(n)$, we consider a predicate over $Comp(\Pi(n)) \times \mathbf{N})$ being, in some way, an invariant of the process of computation in the P system $\Pi(n)$. That is, this predicate will be true for every computation, $\mathcal{C}$, of $\Pi(n)$ and for every natural number. Also, from the truthfulness of the predicate over all the configurations of $\Pi(n)$ must extract relevant information to establish the soundness and completeness of $\Pi(n)$ related to the computing of $f(n)$.

The process of verification of a P system $\Pi(n)$ is based on the analysis of the content of every membrane in every computation that can be obtained in $\Pi(n)$. Given a computation, $\mathcal{C}$, of $\Pi(n)$ we will denote by $\mathcal{C}_k$ the configuration obtained after the execution of k steps in the computation $\mathcal{C}$. In a natural way, a partial function $\mathbf{STEP} : Comp(\Pi) \times \mathbf{N} \times V(\mu_0) - \rightarrow M(A)$ can be defined to assign to every computation, $\mathcal{C}$, of $\Pi(n)$, every natural number, k, and every membrane, i, of the P system $\Pi(n)$, the content of the i-th membrane after the execution of k steps in the computation $\mathcal{C}$. In this moment, if the i-th membrane has been dissolved then the value of $\mathbf{STEP}(\mathcal{C}, k, i)$ is not defined, and we will note $\mathbf{STEP}(\mathcal{C}, k, i) \uparrow$. Otherwise, we denote $\mathbf{STEP}(\mathcal{C}, k, i) \downarrow$. In general, we denote $\mathbf{STEP}(\mathcal{C}, k, i) = \mathcal{C}_k(i)$. Also, we denote by $|\mathcal{C}|$ the length of the computation $\mathcal{C}$, that could be, eventually, infinite.

Definition 4.1 *For every membrane i and every computation $\mathcal{C}$ of $\Pi(n)$, we define $\delta(\mathcal{C}, i) = \min\{m : \mathcal{C}_m(i) \uparrow\}$.*

Since no membrane is dissolved in the initial configuration of a P system, we have $\delta(\mathcal{C}, i) \geq 1$, for every $\mathcal{C} \in Comp(\Pi(n))$ and every membrane i of $\Pi(n)$.

As first result, we will prove that there is only one computation of the P system Π with data input $n = 0$. Moreover, this computation does not halt (and, therefore, it is not successful).

To shorten notation the applicability vector will be expressed with a finite number of components (as many as rules the membrane has). We denote by $\mathbf{0}$ the vector with all null components, irrespectively which is its size.

If $C = (\mu, M)$ is a cell, where $V(\mu) = \{a_1, \ldots, a_n\} \subset \mathbf{N}$ with $a_1 < \cdots < a_n$, we denote $M = (M(a_1), \ldots, M(a_n))$. For simplicity of notation, we represent the multisets by means of the associated word, and $\emptyset$ will be the empty multiset.

Proposition 4.1 *Let $\mathcal{C}$ be a computation of $\Pi(0)$. Then, for every $k \geq 0$, we have $\mathcal{C}_k = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cf^{2^k}))$, and the only applicability matrix over $\mathcal{C}_k$ is $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 0, 2^k))$.*

Proof. Let us prove the result by induction on k. For the base case, $k = 0$, we note that the initial configuration of $\Pi(0)$ is $\mathcal{C}_0 = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cf))$, where μ_0 is the membrane structure associated to Π. Also, $r_5^3 \equiv f \to ff$ is the only rule to be applied and, for this reason, the only applicability matrix of $\mathcal{C}_0$ is $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 0, 1))$.

Assume the result holds for $k \geq 0$. Then $\mathcal{C}_k = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cf^{2^k}))$ and the only applicability matrix over $\mathcal{C}_k$ is $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 0, 2^k))$. Consequently, $\mathcal{C}_{k+1} = \vec{p}(\mathcal{C}_k) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cf^{2^{k+1}}))$. It is clear that the only applicability matrix over $\mathcal{C}_{k+1}$ is $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 0, 2^{k+1}))$. □

Corollary 4.1 *There exists an unique computation of $\Pi(0)$. Furthermore, that computation does not halt.*

Proof. To prove the existence, we consider the initial configuration of $\Pi(0)$, $\mathcal{C}_0 = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cf))$. From Proposition 1 $\mathcal{C}_1 = \vec{p}(\mathcal{C}_0)$, where $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 0, 1))$. In general, if $\mathcal{C}_k = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cf^{2^k}))$ then we consider $\mathcal{C}_{k+1} = \vec{p}(\mathcal{C}_k)$, where $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 0, 2^k))$. The uniqueness of the computation follows directly from Proposition 1.

Moreover, we have proved that $\mathcal{C}$ does not halt, since for each $k \geq 0$ there exists an applicability matrix over $\mathcal{C}_k$ to apply. □

To characterize the successful computation of the P system $\Pi(n)$, with $n \geq 1$, we study what happens when membrane 5 is dissolved. We first determine the content of membrane 5 in each moment of the computation when it is not dissolved.

Proposition 4.2 *Let $n \geq 1$. For every $m \in \mathbf{N}$ ($m \leq n$) and every computation $\mathcal{C}$ of $\Pi(n)$ such that $m < \delta(\mathcal{C}, 5)$ we have $\mathcal{C}_m = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-m}cb^m f^{2^m}))$.*

Proof. Let $n \geq 1$. We consider the P system $\Pi(n)$ with input data n. The proof is by induction on m. For the base case, $m = 0$, it is enough to note that the initial configuration of $\Pi(n)$ is $\mathcal{C}_0 = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^n cf))$.

Let $m \in \mathbf{N}$ be such that $m < n$ and let us suppose that the result holds for m. Let $\mathcal{C}$ be a computation of $\Pi(n)$ such that $m + 1 < \delta(\mathcal{C}, 5)$. Then $m < \delta(\mathcal{C}, 5)$ holds, and consequently, $\mathcal{C}_m = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-m}cb^m f^{2^m}))$. Since $m + 1 < \delta(\mathcal{C}, 5)$ and $m < n$, we conclude that that the only applicability matrix over $\mathcal{C}_m$ is $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 2^m))$, and finally

$$\mathcal{C}_{m+1} = \vec{p}(\mathcal{C}_m) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-m-1} cbb^m f^{2^{m+1}})) = \\ = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-(m+1)} cb^{m+1} f^{2^{m+1}})) \qquad \square$$

Next we present a predicate over the configurations of $\Pi(n)$ to be an invariant along the execution of the P system $\Pi(n)$. Let us consider the formula:

$$\theta(\mathcal{C}, p, n) \equiv \mathcal{C} \in Comp(\Pi(n)) \wedge 1 \leq p \leq n \wedge p = \delta(\mathcal{C}, 5) \rightarrow \mathcal{C} \text{ successful } \wedge O(\mathcal{C}) = p^2$$

Now, the task is to establish a necessary condition for the computation $\mathcal{C}$ of $\Pi(n)$ (with $n \geq 1$) to be successful.

Proposition 4.3 *For every $n \geq 1$ there exists an unique computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) > n$. Furthermore, $\delta(\mathcal{C}, 5) = \infty$ and, therefore, the computation $\mathcal{C}$ does not halt.*

Proof. Let $n \geq 1$. Let us first prove the existence. Let $\mathcal{C}_0$ be the initial configuration of $\Pi(n)$. For every $k \in \mathbf{N}$ such that $1 \leq k \leq n$ we denote $\vec{p}_k = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 2^{k-1}))$ and $D_k = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-k} cb^k f^{2^k}))$.

> **Lemma 1:** *For every k such that $1 \leq k \leq n$ we have $\vec{p}_k$ is an applicability matrix over D_{k-1} and $\mathcal{C}_k = \vec{p}_k(\mathcal{C}_{k-1}) = D_k$.*
>
> *Proof.* The proof is by induction on k.
> To prove the case $k = 1$, since $\vec{p}_1 = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 1))$ is an applicability matrix over D_0, hence $\mathcal{C}_1 = \vec{p}_1(\mathcal{C}_0) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-1} cbf^2)) = D_1$.
> Let k be such that $1 \leq k < n$ and let us suppose that the result holds for k. Since $\vec{p}_k = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 2^{k-1}))$ is an applicability matrix over D_{k-1}, hence $\mathcal{C}_k = \vec{p}_k(\mathcal{C}_{k-1}) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-k} cb^k f^{2^k})) = D_k$. Then $\vec{p}_{k+1} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 2^k))$ is an applicability matrix over D_k. Also, $\mathcal{C}_{k+1} = \vec{p}_{k+1}(\mathcal{C}_k) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-k-1} cb^{k+1} f^{2^{k+1}})) = D_{k+1}$. □

From lemma 1 we deduce that $\mathcal{C}_n = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cb^n f^{2^n})) = D_n$. For every $q \in \mathbf{N}$ such that $q \geq 1$, we denote $\vec{p}_{n+q} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 2^{n+q-1}))$ and $D_{n+q} = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cb^n f^{2^{n+q}}))$.

> **Lemma 2:** *For every $q \geq 1$ we have $\vec{p}_{n+q}$ is an applicability matrix over D_{n+q-1} and, also, $\mathcal{C}_{n+q} = \vec{p}_{n+q}(\mathcal{C}_{n+q-1}) = D_{n+q}$.*
>
> *Proof.* By induction on q.
> To prove the base case, $q = 1$, it suffices to have in mind that $\vec{p}_{n+1} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 0, 2^n))$ is an applicability matrix over D_n. Also, $\mathcal{C}_{n+1} = \vec{p}_{n+1}(\mathcal{C}_n) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cb^n f^{2^{n+1}})) = D_{n+1}$.
> Let $q \geq 1$ and let us suppose the result holds for q. We have $\mathcal{C}_{n+q} = D_{n+q} = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cb^n f^{2^{n+q}}))$. Then $\vec{p}_{n+q+q} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 2^{n+q}))$ is an applicability matrix over D_{n+q}. Also, $\mathcal{C}_{n+q+q} = \vec{p}_{n+q+q}(\mathcal{C}_{n+q}) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, cb^n f^{2^{n+q+1}})) = D_{n+q+1}$. □

The computation $\mathcal{C}$ of $\Pi(n)$ built as above verifies that $\delta(\mathcal{C}_n, 5) = \infty$ and, therefore, it does not halt.

Finally we establish the uniqueness. For it, let us suppose that $\mathcal{C}, \mathcal{C}'$ are computations of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) > n$ and $\delta(\mathcal{C}', 5) > n$. From Proposition 2 we deduce that for every $m \in \mathbf{N}$ such that $m \leq n$ is is verified that $\mathcal{C}_m = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-m} c b^m f^{2^m})) = \mathcal{C}'_m$. Since $\mathcal{C}_n = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, c b^n f^{2^n}))$, it results for every $q \geq 1$ that $\mathcal{C}_{n+q} = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, c b^n f^{2^{n+q}})) = \mathcal{C}'_{n+q}$. Therefore, the computation $\mathcal{C}$ does not halt. □

Corollary 4.2 *Let $n \geq 1$. Let $\mathcal{C}$ be a successful computation of $\Pi(n)$. Then $\forall k$ $(k < \delta(\mathcal{C}, 5) \rightarrow k < n)$.*

Proof. Let $n \geq 1$. Let $\mathcal{C}$ be a computation of $\Pi(n)$ such that there exists $k \geq n$ verifying that $k < \delta(\mathcal{C}, 5)$. Then, $n < \delta(\mathcal{C}_n, 5)$ and, therefore, from Proposition 4.3 we deduce that the computation $\mathcal{C}$ does not halt, which is impossible. □

Next, let us see that for every $n \geq 1$ a successful computation of $\Pi(n)$ whose output is 1 can be built.

Proposition 4.4 *Let $n \geq 1$. There exists an unique computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = 1$. Furthermore, this computation is successful, its length is 3 and its output is $O(\mathcal{C}) = 1$.*

Proof. Let $n \geq 1$. To build a computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = 1$, we note that $\mathcal{C}_0(5) \downarrow$, and therefore, in the first step of the computation $\mathcal{C}$ the rules $r_2^5 \equiv ac \rightarrow bc\delta$ and $r_3^5 \equiv f \rightarrow ff$ must be applied (in a maximal manner). That is, to get $\delta(\mathcal{C}, 5) = 1$, the only applicability matrix that can be applied over $\mathcal{C}_0$ is $\vec{p}_1 = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 1))$. Hence, $\mathcal{C}_1 = \vec{p}_1(\mathcal{C}_0) = (\mu_1, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-1} c b f^2))$, where $\mu_1 = (1, ((1, 2), (2, 3, 4), (3, 2), (4, 2)))$. But, the only applicability matrix over $\mathcal{C}_1$ is $\vec{p}_2 = (\mathbf{0}, \mathbf{0}, (1), \mathbf{0})$. Therefore, $\mathcal{C}_2 = \vec{p}_1(\mathcal{C}_1) = (\mu_2, (\emptyset, a^{n-1} c b g, \emptyset))$, where $\mu_2 = (1, ((1, 2), (2, 1, 4), (4, 2)))$. Then, the only applicability matrix over $\mathcal{C}_2$ is $\vec{p}_2 = (\mathbf{0}, (1, 0, 1), \mathbf{0})$. Hence, $\mathcal{C}_3 = \vec{p}_1(\mathcal{C}_2) = (\mu_3, (a^n c b, e))$, where $\mu_3 = (1, ((1, 4), (4, 1)))$.

Having in mind that membranes 1 and 4 have no rules, it results that the configuration $\mathcal{C}_3$ halts. Moreover, we deduce that $\mathcal{C}_3$ is a successful configuration, because $i_0 = 4 \in \mu_3$. Therefore, $|C| = 3$ and the output of the computation $\mathcal{C}$ is $O(\mathcal{C}) = |\mathcal{C}_3(4)| = |\{e\}| = 1$.

The uniqueness of such a computation $\mathcal{C}$ of $\Pi(n)$ verifying $\mathcal{C}_1(5) \uparrow$ follows from the above construction. □

We now proceed to prove that the critical point of the computations of the P system $\Pi(n)$, with $n \geq 1$, is the dissolution of membrane 5.

Proposition 4.5 *Let $n \geq 1$. Let m be such that $2 \leq m \leq n$. For every computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = m$ we have:*

1. *$\mathcal{C}_m = (\mu', (\emptyset, \emptyset, a^{n-m} c b^m f^{2^m}, \emptyset))$, where μ' is the rooted tree:*
$$(1, ((1, 2), (2, 1, 3, 4), (3, 2), (4, 2)))$$
2. *For every k $(0 \leq k \leq m - 1)$, $\mathcal{C}_{m+1+k} = (\mu'', (\emptyset, a^{n-m} c b^m g^{2^{m-1-k}}, e^{km}))$, where $\mu'' = (1, ((1, 2), (2, 1, 4), (4, 2)))$.*
3. *$\mathcal{C}_{2m+1} = (\mu''', (a^{n-m+1} c b^m, e^{m^2}))$, where $\mu''' = (1, ((1, 4), (4, 1)))$.*

4. *The computation $\mathcal{C}$ is successful, its length is $|\mathcal{C}| = 2m+1$, and its output is $O(\mathcal{C}) = m^2$.*

Proof. (1) The proof is by induction on m. To prove the base case, $m = 2 \leq n$, we consider a computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = 2$. Since $1 < \delta(\mathcal{C}, 5)$, from Proposition 4.2 it follows that $\mathcal{C}_1 = (\mu_0, (\emptyset, \emptyset, \emptyset, a^{n-1}cbf^2))$. Since $\delta(\mathcal{C}, 5) = 2 \leq n$, it results that in the second step of the computation $\mathcal{C}$ the rules $r_2^5 \equiv ac \to bc\delta$ and $r_3^5 \equiv f \to ff$ must be applied (in a maximal manner). Therefore, $\mathcal{C}_2 = \vec{p}(\mathcal{C}_1) = (\mu', (\emptyset, \emptyset, a^{n-2}cb^2 f^{2^2}, \emptyset))$, where $\mu' = (1, ((1,2),(2,1,3,4),(3,2),(4,2)))$ and $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0,1,2))$.

Let $m \in \mathbf{N}$ such that $2 \leq m < n$ and let us suppose that the result holds for m. Let $\mathcal{C}$ be a computation of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = m+1$. Since $m < \delta(\mathcal{C}, 5)$ and $m < n$, from Proposition 4.2 it follows that $\mathcal{C}_m = \vec{p}(\mathcal{C} - 1) = (\mu_0, (\emptyset, \emptyset, \emptyset, a^{n-m}cb^m f^{2^m}))$. Since $\delta(\mathcal{C}, 5) = m+1$, in the $(m+1)$–th step of the computation $\mathcal{C}$, the membrane 5 must be dissolved. Hence, the rules $r_2^5 \equiv ac \to bc\delta$ and $r_3^5 \equiv f \to ff$ must be applied (in a maximal manner). Therefore,

$$\mathcal{C}_{m+1} = \vec{p}(\mathcal{C}_m) = (\mu', (\emptyset, \emptyset, a^{n-m-1}cb^{m+1}f^{2^{m+1}}, \emptyset))$$

where $\mu' = (1, ((1,2),(2,1,3,4),(3,2),(4,2)))$ and $\vec{p} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0,1,2^m))$.

(2) Let $n \geq 1$. Let $m \in \mathbf{N}$ such that $2 \leq m \leq n$. Let $\mathcal{C}$ be a computation of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = m$. Let us first prove that for every k such that $0 \leq k \leq m-1$ we have $\mathcal{C}_{m+1+k} = (\mu'', (\emptyset, a^{n-m}cb^m g^{2^{m-1-k}}, e^{km}))$, where $\mu'' = (1, ((1,2),(2,1,4),(4,2)))$. The proof is by induction on k. To prove the base case, $k = 0$, from (1) it follows that $\mathcal{C}_m = (\mu', (\emptyset, \emptyset, a^{n-m}cb^m f^{2^m}, \emptyset))$, where $\mu' = (1, ((1,2),(2,1,3,4),(3,2),(4,2)))$, hence, the only applicability matrix over $\mathcal{C}_m$ is $\vec{p} = (\mathbf{0}, \mathbf{0}, (2^{m-1}), \mathbf{0})$, and finally

$$\mathcal{C}_{m+1} = \vec{p}(\mathcal{C}_m) = (\mu'', (\emptyset, a^{n-m}cb^m g^{2^{m-1}}, \emptyset))$$

where $\mu'' = (1, ((1,2),(2,1,4),(4,2)))$.

For each $k \geq 0$ such that $k < n-1$ assume that the result holds for k. The induction hypothesis states that $\mathcal{C}_{m+1+k} = (\mu'', (\emptyset, a^{n-m}cb^m g^{2^{m-1-k}}, e^{km}))$, where $\mu'' = (1, ((1,2),(2,1,4),(4,2)))$. Since $m-1-k > 0$ and $m > 0$, the only applicability matrix over $\mathcal{C}_{m+1+k}$ is $\vec{p} = (\mathbf{0}, (m, 2^{m-1-k-1}, 0), \mathbf{0})$. Hence

$$\begin{aligned}\mathcal{C}_{m+1+k+1} = \vec{p}(\mathcal{C}_{m+1+k}) &= (\mu'', (\emptyset, a^{n-m}cb^m g^{2^{m-1-k-1}}, e^m e^{km})) = \\ &= (\mu'', (\emptyset, a^{n-m}cb^m g^{2^{m-1-(k+1)}}, e^{(k+1)m}))\end{aligned}$$

(3) Let $n \geq 1$. Let $m \in \mathbf{N}$ such that $2 \leq m \leq n$. Let $\mathcal{C}$ be a computation of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = m$. From (2), $\mathcal{C}_{2m} = (\mu'', (\emptyset, a^{n-m}cb^m g, e^{(m-1)m}))$, where $\mu'' = (1, ((1,2),(2,1,4),(4,2)))$. Hence, the only applicability matrix over $\mathcal{C}_{2m}$ is $\vec{p} = (\mathbf{0}, (m, 0, 1), \mathbf{0})$, and, consequently,

$$\begin{aligned}\mathcal{C}_{2m+1} = \vec{p}(\mathcal{C}_{2m}) &= (\mu''', (\emptyset, a^{n-m+1}cb^m, e^m e^{(m-1)m})) = \\ &= (\mu''', (\emptyset, a^{n-m+1}cb^m, e^{m^2}))\end{aligned}$$

where $\mu''' = (1, ((1,4),(4,1)))$.

(4) Let $n \geq 1$. Let $m \in \mathbf{N}$ such that $2 \leq m \leq n$. Let $\mathcal{C}$ be a computation of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = m$. From (3), we deduce that $\mathcal{C}_{2m+1}$ is a halting configuration, since the only non empty membrane are 1 and 4, and both of them have no rules. Therefore, $|\mathcal{C}| = 2m+1$. Moreover, the computation $\mathcal{C}$ is successful, and its output is $O(\mathcal{C}) = |\mathcal{C}_{2m+1}(4)| = |\{\{e^{m^2}\}\}| = m^2$. □

Corollary 4.3 *Let $n \geq 1$. For every p such that $1 \leq p \leq n$ there exists at most one computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = p$.*

Proof. Let $n \geq 1$. If $p = 1$, from Proposition 4.4 it follows that there exists an unique computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = p$.

Let p be such that $2 \leq p \leq n$. Let $\mathcal{C}, \mathcal{C}'$ be computations of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = \delta(\mathcal{C}', 5) = p$. From (4) in Proposition 4.5 we deduce that $|\mathcal{C}| = |\mathcal{C}'| = 2p + 1$. Also, Proposition 4.2 shows that $\forall k\ (0 \leq k < p \rightarrow \mathcal{C}_k = \mathcal{C}'_k)$. Let us see that $\forall k\ (p \leq k \leq 2p + 1 \rightarrow \mathcal{C}_k = \mathcal{C}'_k)$.

- The case $k = p$ follows from (1) in Proposition 4.5.
- The case $p + 1 \leq k \leq 2p$ follows from (2) in Proposition 4.5.
- The case $k = 2p + 1$ follows from (3) in Proposition 4.5.

Consequently, $\mathcal{C} = \mathcal{C}'$. □

Now we can characterize the successful computations of Π through the moment when membrane 5 is dissolved.

Corollary 4.4 *Let $n \geq 1$. Let $\mathcal{C}$ be a computation of $\Pi(n)$. The following conditions are equivalent:*

(a) $\mathcal{C}$ *is a successful computation.*

(b) $1 \leq \delta(\mathcal{C}, 5) \leq n$.

(c) $1 \leq \delta(\mathcal{C}, 5) \leq n \wedge |\mathcal{C}| = 2\delta(\mathcal{C}, 5) + 1 \wedge O(\mathcal{C}) = \delta(\mathcal{C}, 5)^2$.

Proof. Let $n \geq 1$. Let $\mathcal{C}$ be a successful computation of $\Pi(n)$. From proposition 4.3 we deduce that $\delta(\mathcal{C}, 5) \leq n$.

Let $\mathcal{C}$ a successful computation of $\Pi(n)$ such that $1 \leq \delta(\mathcal{C}, 5) \leq n$. If $\delta(\mathcal{C}, 5) = 1$, then from Proposition 4.5 we deduce that $|\mathcal{C}| = 3$ and $O(\mathcal{C}) = 1$. If $2 \leq \delta(\mathcal{C}, 5) \leq n$, then from (4) of Proposition 4.5 we deduce that $|\mathcal{C}| = 2\delta(\mathcal{C}, 5) + 1 \wedge O(\mathcal{C}) = \delta(\mathcal{C}, 5)^2$.

Let $\mathcal{C}$ be a successful computation of $\Pi(n)$ such that $1 \leq \delta(\mathcal{C}, 5) \leq n \wedge |\mathcal{C}| = 2\delta(\mathcal{C}, 5) + 1 \wedge O(\mathcal{C}) = \delta(\mathcal{C}, 5)^2$. If $\delta(\mathcal{C}, 5) = 1$ then from Proposition 4.4 it is deduced that $\mathcal{C}$ is successful. If $2 \leq \delta(\mathcal{C}, 5) \leq n$ then from (4) of Proposition 4.5 we deduce that the computation $\mathcal{C}$ is successful. □

Corollary 4.5 *For every $n \geq 1$ there exists an unique non halting computation of $\Pi(n)$.*

Proof. Let $n \geq 1$. If $\mathcal{C}$ is a computation of $\Pi(n)$ that it does not halt, than it is not successful and, hence, $\delta(\mathcal{C}, 5) > n$. From Proposition 4.3 we deduce that it is unique. □

5 Soundness and Completeness of the computing P system Π

To establish that the computing P system Π described in section 3 actually computes the partial function $f : \mathbf{N} - \rightarrow P(\mathbf{N})$ defined by

$$f(n) = \begin{cases} \uparrow & \text{if } n = 0 \\ \{1^2, 2^2, \ldots, n^2\} & \text{if } n \neq 0 \end{cases}$$

we must to prove that:

(a) No computation of $\Pi(0)$ is successful.

(b) For every $n \geq 1$ we have:

- For every successful computation, $\mathcal{C}$, of $\Pi(n)$ there is $p \in \mathbf{N}$ such that $1 \leq p \leq n$ and the output of the computation $\mathcal{C}$ is p^2 (*soundness*).
- For every $p \in \mathbf{N}$ such that $1 \leq p \leq n$, there exists at least one successful computation, $\mathcal{C}$, of $\Pi(n)$ whith the output $O(\mathcal{C}) = p^2$ (*completeness*).

In other words, we must to prove that no computation of $\Pi(0)$ is a halting one and, also, that $\forall n \geq 1$ $(O(\Pi(n)) \subseteq f(n))$ (*soundness*) and that $\forall n \geq 1$ $(f(n) \subseteq O(\Pi(n)))$ (*completeness*).

Theorem 5.1 *(***Soundness***) For every $n \geq 1$ and every successful computation, $\mathcal{C}$, of $\Pi(n)$, there is $p \in \mathbf{N}$ such that $1 \leq p \leq n$ and $O(\mathcal{C}) = p^2$. That is to say, $\forall n \geq 1$ $(O(\Pi(n)) \subseteq f(n))$.*

Proof. Let $n \geq 1$. Let $\mathcal{C}$ be a successful computation of $\Pi(n)$. From Corollary 4.5 it follows that $1 \leq \delta(\mathcal{C}, 5) \leq n \wedge |\mathcal{C}| = 2\delta(\mathcal{C}, 5) + 1 \wedge O(\mathcal{C}) = \delta(\mathcal{C}, 5)^2$. If $\delta(\mathcal{C}, 5) = 1$, Proposition 4.4 shows that $O(\mathcal{C}) = 1$, and if $\delta(\mathcal{C}, 5) = m$ $(2 \leq m \leq n)$, from (4) in Proposition 4.4 it follows that $O(\mathcal{C}) = m^2$. □

To establish the completeness of Π we consider the following formula

$$\varphi(n,p) \equiv \exists \mathcal{C} \in Comp(\Pi(n))\ (p = \delta(\mathcal{C}, 5))$$

Proposition 5.1 *Let $n \geq 1$. For every $p \in \mathbf{N}$ such that $1 \leq p \leq n$ there exists an unique computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = p$.*

Proof. Let $n \geq 1$. We begin by proving the existence by induction on p. The base case, $p = 1$, follows directly from Proposition 4.4. For each p be such that $1 \leq p < n$ assume that the result holds for p. Let $\mathcal{C}$ a computation of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = p$. Since $0 \leq p-1 < p = \delta(\mathcal{C}, 5)$, from Proposition 4.2 it follows that $\mathcal{C}_{p-1} = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-p+1} c b^{p-1} f^{2^{p_1}}))$. Since $\vec{p}_1 = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (1, 0, 2^{p-1}))$ is an applicability matrix over $\mathcal{C}_{p-1}$, we define

$$\mathcal{C}'_p = \vec{p}_1(\mathcal{C}_{p-1}) = (\mu_0, (\emptyset, \emptyset, \emptyset, \emptyset, a^{n-p} c b^p f^{2^p}))$$

Since $\vec{p}_2 = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, (0, 1, 2^p))$ is an applicability matrix over $\mathcal{C}'_p$, we define $\mathcal{C}'_{p+1} = \vec{p}_2(\mathcal{C}'_p) = (\mu', (\emptyset, \emptyset, \emptyset, a^{n-p-1} c b^{p+1} f^{2^{p+1}}, \emptyset))$, where the membrane structure $\mu' = (1, ((1,2), (2,1,3,4), (3,2), (4,2)))$. Therefore, the computation of $\Pi(n)$: $\mathcal{C}_0 \Rightarrow_\Pi \mathcal{C}_1 \Rightarrow_\Pi \ldots \Rightarrow_\Pi \mathcal{C}'_p \Rightarrow_\Pi \mathcal{C}'_{p+1} \Rightarrow_\Pi \ldots$, verifies that $\delta(\mathcal{C}, 5) = p + 1$, which proves the result holds for $p + 1$.

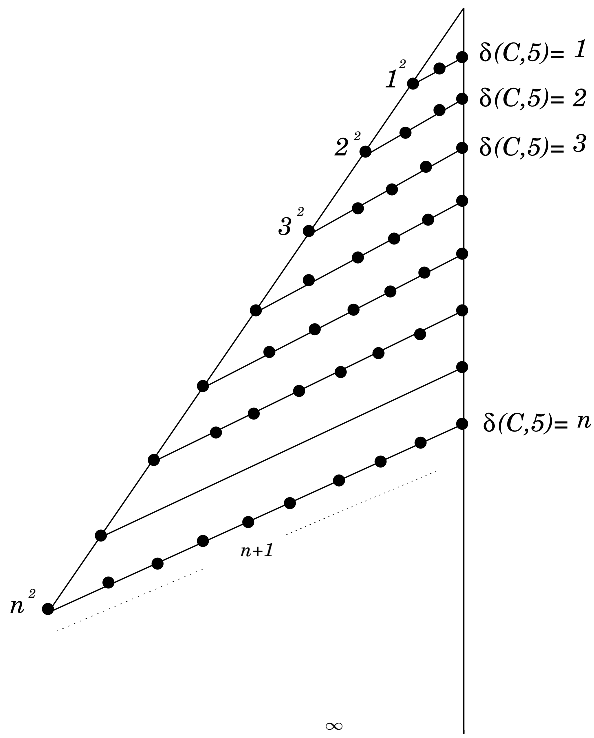
The uniqueness of such a computation $\mathcal{C}$, of $\Pi(n)$ verifying $\delta(\mathcal{C}, 5) = p$, follows from Corollary 4.3. □

Corollary 5.1 *For every $n \geq 1$ it is verified that $\forall p\ (1 \leq p \leq n \rightarrow \varphi(n,p))$.*

Theorem 5.2 *(***Completeness***) Let $n \geq 1$. For every $p \in \mathbf{N}$ such that $1 \leq p \leq n$, there exists a computation, $\mathcal{C}$, of $\Pi(n)$ such that $\mathcal{C}$ is successful and its output is $O(\mathcal{C}) = p^2$. That is, $\forall n \geq 1$ $(f(n) \subseteq O(\Pi(n)))$.*

Proof. Let $n \geq 1$. Let p be such that $1 \leq p \leq n$. From Proposition 5.1 it follows that there is an unique computation, $\mathcal{C}$, of $\Pi(n)$ such that $\delta(\mathcal{C}, 5) = p$. If $p = 1$ the result follows from Proposition 4.4. If $2 \leq p \leq n$ the result follows from (4) of Proposition 4.5. □

Summarizing, we can describe the set of all computations of the P system $\Pi(n)$, for every $n \geq 1$, as follows:



Note that for every $n \geq 1$, the P system $\Pi(n)$ has exactly $n + 1$ computations, where only one does not halt (and, therefore, it is not successful).

6 Conclusions

The formal verification of mechanical procedures in a computing model use to be a complex task. If the mechanical procedures of the model are not described through an imperative language then this task gets harder. As it is known, the P systems are, basically, of a procedural kind and, consequently, the task to give formal verification of a P system is very complicated.

In this work the problem of formal verification of a computing P system to compute a partial function that for every $n \geq 1$ returns the set $\{1^2, \ldots, n^2\}$ has been studied. For it, the *critical points* of the computations of the system (the instants where some important fact happens) are established and characteristic properties of successful computations are obtained.

The study of formal verification of P systems can represent an important step through the treatment of them with reasoning automated systems. Also, together with the obtained formalization in [3], this paper can be useful for a possible implementation of P systems into conventional electronic computers.

Acknowledgement

Support for this research through the project TIC2002-04220-C03-01 of the Ministerio de Ciencia y Tecnología of Spain, cofinanced by FEDER funds, is gratefully acknowledged.

References

[1] Gh. Păun, Computing with membranes, *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for Computer Science-TUCS Report* No 208, 1998 (www.tucs.fi).

[2] Gh. Păun; G. Rozenberg, A guide to membrane computing, *Theoretical Computer Science*, 287 (2002), 73–100.

[3] M.J. Pérez–Jiménez; F. Sancho-Caparrini. A formalization of transition P systems. *Fundamenta Informaticae*, vol. 49 (2002), 261–272.

[4] M.J. Pérez–Jiménez; F. Sancho-Caparrini. Verifying a P system generating squares. *Romanian Journal of Information Science and Technology*, vol. 5, núm, 2–3 (2002), 181–191.